

When coding meets ranking: A joint framework based on local learning

Jim Jing-Yan Wang

Computer, Electrical and Mathematical Sciences and Engineering Division

King Abdullah University of Science and Technology (KAUST)

Thuwal 23955-6900, Saudi Arabia

jimjywang@gmail.com

Abstract

Sparse coding, which represents a data point as a sparse reconstruction code with regard to a dictionary, has been a popular data representation method. Meanwhile, in database retrieval problems, learn the ranking scores from data points plays an important role. Up to now, these two methods have always been used individually, assuming that data coding and ranking are two independent and irrelevant problems. However, is there any internal relationship between sparse coding and ranking score learning? If yes, how to explore this internal relationship? In this paper, we try to answer these questions by developing the first joint sparse coding and ranking score learning algorithm. To explore the local distribution in the sparse code space, and also to bridge coding and ranking problems, we assume that in the neighborhood of each data points, the ranking scores can be approximated from the corresponding sparse codes by a local linear function. By considering the local approximation error of ranking scores, reconstruction error and sparsity of sparse coding, and the query information provided by the user, we construct an unified objective function for learning of sparse codes, dictionary and rankings scores. An iterative algorithm is developed to optimize the objective function to jointly learn the sparse codes, dictionary and rankings scores.

1. Introduction

Sparse coding is a popular data representation method [7]. It tries to reconstruct a given data point as a linear combination of some basic elements in a dictionary, which are called codewords. The linear combination coefficients are imposed to be sparse, e.g., most of the combination coefficients are zeros. The linear combination coefficient vector of a data point can be used as its new representation, and we call it sparse code due to its sparsity. Because of its ability to explore the latent part-based nature of the data, it has been used widely to represent data in pattern classification

problems. Many sparse coding algorithms are proposed to the dictionary and sparse codes [3, 1, 10, 5, 6].

Meanwhile, in nearest neighbor-based classification and content based database retrieval problems, the data points are usually ranked according to their similarity measures to the queries. The similarity measures are referred as ranking scores. Recently, methods to learn the ranking scores from the data points are proposed and show its power in retrieval problems [2]. By considering both the query information provided by the users and the distribution information of the data points, efficient algorithms are developed to learn the ranking scores [9, 8].

It is possible to use both sparse coding and ranking score learning technologies to boost the performance of nearest neighbor searching. We may firstly map the data points to the sparse codes using a sparse coding algorithm, and then learn the ranking scores in the sparse code space. However, this strategy uses sparse coding and ranking methods independently and assume that they are two irrelevant problems. In this paper, we ask the following two questions about sparse coding and ranking score learning:

1. Is there any internal relationship between sparse coding and ranking score learning?
2. If yes, how can we explore it to boost both the data representation and ranking?

To answer these two questions, we propose to learn the sparse codes and ranking scores jointly to explore their internal relationship. Actually, in [4], Mairal et al. proposed to learn sparse codes, a dictionary and a classifier jointly to explore the internal relationship between sparse coding and classification. However, up to now, there is no work conducted to consider both sparse coding and ranking problems jointly.

To this end, we proposed to perform sparse coding to all the data points and use the query information provided by the user to regularize the learning of the ranking scores. More importantly, we to bridge the learning of sparse codes

and ranking scores, and also to utilize the local distribution of the data points, we assume that in a local neighborhood of each data points, the ranking scores can be approximated from the sparse codes using a local linear function. By considering the reconstruction error and sparsity of sparse coding problems, the local approximation error and the complexity of local ranking score approximation, and the query information regularization problems simultaneously, we construct an unified objective function for learning of the sparse codes, dictionary and ranking scores. By optimizing this objective function, sparse codes and ranking scores can regularize the learning of each other, and thus the internal relationship can be explored. An iterative algorithm is developed to optimize the objective function with regard to the sparse codes, dictionary and ranking scores, using the alternative optimization strategy.

2. Proposed Method

In this section, we will introduce the proposed unified sparse coding and ranking score learning method.

2.1. Problem Formulation

We assume we have a data set of n data points, denoted as $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d -dimensional feature vector of the i -th data point. In this data set, most of the data points are from a given database, while only one or a few data points are provided by the user, which are named as queries. The problem of data retrieval is to return the some data points from the database which are most similar to the queries. To indicate the query data points, we define a query indicator vector $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_n] \in \{1, 0\}^n$, where $\lambda_i = 1$ if \mathbf{x}_i is a query, and 0 otherwise. To this end, ranking scores are learned for the data points as similarities to the queries so that the data points can be ranked according to the ranking scores, and the top ranked data points are returned as retrieval results. The ranking scores of the data points in \mathcal{X} are organized in a ranking score vector $\mathbf{f} = [f_1, \dots, f_n]^\top \in \mathbb{R}^n$, where f_i is the ranking score for the i -th data point. To learn the ranking score, we represent the data points as sparse codes of a dictionary first, and then learn the ranking scores from the sparse codes and query information. The following problems are considered to construct a unified objective function to learn both the sparse codes and the ranking scores.

- **Sparse coding** The sparse coding problem aims to learn a dictionary with m codewords $\{\mathbf{d}_l\}_{l=1}^m$, and reconstruct a data point \mathbf{x}_i as a sparse linear combination of the codewords,

$$\mathbf{x}_i \approx \sum_{l=1}^m \mathbf{d}_l s_{il} = D\mathbf{s}_i \quad (1)$$

where $D = [\mathbf{d}_1, \dots, \mathbf{d}_m] \in \mathbb{R}^{d \times m}$ is the dictionary matrix, $\mathbf{d}_l \in \mathbb{R}^d$ is the l -th codeword, and $\mathbf{s}_i = [s_{i1}, \dots, s_{im}]^\top \in \mathbb{R}^m$ is the sparse code of \mathbf{x}_i . To learn the dictionary and the sparse codes of the data points, the following minimization problem is considered,

$$\begin{aligned} \min_{D, \{\mathbf{s}_i\}_{i=1}^n} \sum_{i=1}^n \left(\|\mathbf{x}_i - D\mathbf{s}_i\|_2^2 + \alpha \|\mathbf{s}_i\|_1 \right), \\ \text{s.t. } \|\mathbf{d}_l\|_2^2 \leq C, l = 1, \dots, m, \end{aligned} \quad (2)$$

where $\|\mathbf{x}_i - D\mathbf{s}_i\|_2^2$ is the reconstruction error of the i -th data point measured by the squared ℓ_2 norm distance, $\|\mathbf{s}_i\|_1$ is a ℓ_1 norm based sparsity measure of the sparse code \mathbf{s}_i , and α is a tradeoff parameter. By solving this problem, the data points are represented as the corresponding sparse codes. We will use the sparse codes to predict their ranking scores.

- **Local ranking score learning** To unitize the local structure of the sparse code space, we propose to learn a local linear function for the neighborhood of each data point to approximate the ranking scores. The k -nearest neighboring data points of \mathbf{x}_i is denoted as \mathcal{N}_i . We propose to learn a linear function $h_i(\mathbf{s}_j)$ to approximate the ranking scores $f_j|_{j:\mathbf{x}_j \in \mathcal{N}_i}$ of data points in this neighborhood from there sparse codes $\mathbf{s}_j|_{j:\mathbf{x}_j \in \mathcal{N}_i}$,

$$f_j \approx h_i(\mathbf{s}_j) = \mathbf{w}_i^\top \mathbf{s}_j \quad (3)$$

where $\mathbf{w}_i \in \mathbb{R}^m$ is the parameter vector of the linear function of the \mathcal{N}_i . To learn \mathbf{w}_i , we propose the following minimization problem for each \mathcal{N}_i

$$\min_{\{\mathbf{s}_j, f_j\}_{j:\mathbf{x}_j \in \mathcal{N}_i}, \mathbf{w}_i} \sum_{j:\mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2 + \beta \|\mathbf{w}_i\|_2^2 \quad (4)$$

where $\|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2$ is the approximation error of ranking scores measured by squared ℓ_2 norm, $\|\mathbf{w}_i\|_2^2$ is a square ℓ_2 norm based regularization term used control the complex of the local linear function, and β is a tradeoff parameter. An overall problem is obtained by summing up the local minimization problems over all the data points,

$$\min_{\{\mathbf{s}_i, f_i, \mathbf{w}_i\}_{i=1}^n} \sum_{i=1}^n \left(\sum_{j:\mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2 + \beta \|\mathbf{w}_i\|_2^2 \right). \quad (5)$$

Please note that not only the local function parameters $\mathbf{w}_i|_{i=1}^n$ are to be solved, but also the sparse codes and ranking scores.

- **Query regularization** To unitize the query information provided by the users, we also regularize the learning of the ranking scores with the query indicator. If a

data point is a query, its ranking score should be large since it is similar to itself. Thus we define a large value constant y and force the ranking scores of the queries to be close to it. The following minimization problem is obtained,

$$\min_{\{f_i\}_{i=1}^n} \sum_{i=1}^n \|f_i - y\|_2^2 \lambda_i \quad (6)$$

In this problem, when a data point \mathbf{x}_i is a query ($\lambda_i = 1$), we minimize squared ℓ_2 norm distance between its ranking score f_i and the large value y .

The overall optimization problem is obtained by combining the problems in (2), (5), and (6),

$$\begin{aligned} \min_{D, \{\mathbf{s}_i, f_i, \mathbf{w}_i\}_{i=1}^n} \{ & O(D, \mathbf{s}_i)_{i=1}^n, f_i)_{i=1}^n, \mathbf{w}_i)_{i=1}^n = \\ & \sum_{i=1}^n \left(\|\mathbf{x}_i - D\mathbf{s}_i\|_2^2 + \alpha \|\mathbf{s}_i\|_1 \right) \\ & + \gamma \sum_{i=1}^n \left(\sum_{j: \mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2 + \beta \|\mathbf{w}_i\|_2^2 \right) \\ & + \delta \sum_{i=1}^n \|f_i - y\|_2^2 \lambda_i \} \\ \text{s.t. } & \|\mathbf{d}_l\|_2^2 \leq C. \end{aligned} \quad (7)$$

where γ and δ are tradeoff parameters. Please note that in this problem, we need to solve a dictionary D , the corresponding sparse codes $\{\mathbf{s}_i\}_{i=1}^n$, the ranking scores $\{f_i\}_{i=1}^n$, and the local linear ranking score predictor parameters $\{\mathbf{w}_i\}_{i=1}^n$ of the data points. The learning of these variables are unified in a single optimization problem, and thus the learning of them are regularized by each other. This is the critical difference between the proposed method and the traditional independent sparse coding and ranking score learning algorithms which ignores the inner connection between them.

2.2. Optimization

Directly solving this problem is difficult, and we adapt the alternative optimization strategy to solve it. The ranking scores, sparse codes and dictionary are updated in an iterative algorithm. In each iteration, one of them is solved while the others are fixed as previous results, then their roles are switched. The iterations are repeated until convergency.

2.2.1 Solving ranking scores

When the ranking scores $\{f_i\}_{i=1}^n$ are been solved, we fix D and $\{\mathbf{s}_i\}_{i=1}^n$, remove the objective terms irrelevant to ranking scores from (7), and obtain the following problem,

$$\begin{aligned} \min_{\{f_i, \mathbf{w}_i\}_{i=1}^n} \{ & \gamma \sum_{i=1}^n \left(\sum_{j: \mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2 + \beta \|\mathbf{w}_i\|_2^2 \right) \\ & + \delta \sum_{i=1}^n \|f_i - y\|_2^2 \lambda_i \\ & = \sum_{i=1}^n g(S_i, \mathbf{f}_i, \mathbf{w}_i) + \delta \sum_{i=1}^n \|f_i - y\|_2^2 \lambda_i \} \end{aligned} \quad (8)$$

where $g(S_i, \mathbf{f}_i, \mathbf{w}_i) = \sum_{j: \mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2 + \beta \|\mathbf{w}_i\|_2^2$ is define the local objective for each local ranking score learning problem of \mathbf{x}_i . To rewrite it in matrix form, we define a local ranking score vector for each \mathcal{N}_i as $\mathbf{f}_i = [f_{i1}, \dots, f_{ik}] \in \mathbb{R}^k$, where f_{ij} is the ranking score of the j -th nearest neighbor point of \mathbf{x}_i . Similarly, we define a local sparse code matrix for each \mathcal{N}_i as $S_i = [\mathbf{s}_{i1}, \dots, \mathbf{s}_{ik}] \in \mathbb{R}^{m \times k}$, where \mathbf{s}_{ij} is the sparse code of the j -th nearest neighbor point of \mathbf{x}_i . In this way, we rewrite $g(S_i, \mathbf{f}_i, \mathbf{w}_i)$ as

$$\begin{aligned} g(S_i, \mathbf{f}_i, \mathbf{w}_i) &= \sum_{j: \mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2 + \beta \|\mathbf{w}_i\|_2^2 \\ &= \|\mathbf{f}_i - \mathbf{w}_i^\top S_i\|_2^2 + \beta \|\mathbf{w}_i\|_2^2. \end{aligned} \quad (9)$$

Please note that the objective function of (8) is composed of the local objective functions of all data points, thus this local objective function is to be minimized. To minimize it, we set its partial derivative with regard to \mathbf{w}_i to zero,

$$\frac{\partial g}{\partial \mathbf{w}_i} = -2S_i \mathbf{f}_i^\top + 2S_i S_i^\top \mathbf{w}_i + 2\beta \mathbf{w}_i = 0, \quad (10)$$

and obtain the following solution for \mathbf{w}_i

$$\mathbf{w}_i = (S_i S_i^\top + \beta I)^{-1} S_i \mathbf{f}_i^\top = \Phi_i \mathbf{f}_i^\top, \quad (11)$$

where

$$\Phi_i = (S_i S_i^\top + \beta I)^{-1} S_i \in \mathbb{R}^{m \times k} \quad (12)$$

By substituting it to (9), we can eliminate \mathbf{w}_i from (9) and rewrite it as

$$\begin{aligned} g(S_i, \mathbf{f}_i) &= \left\| \mathbf{f}_i - \left(\Phi_i \mathbf{f}_i^\top \right)^\top S_i \right\|_2^2 + \beta \left\| \Phi_i \mathbf{f}_i^\top \right\|_2^2 \\ &= \left\| \mathbf{f}_i (I - \Phi_i^\top S_i) \right\|_2^2 + \beta \left\| \mathbf{f}_i \Phi_i^\top \right\|_2^2 \\ &= \mathbf{f}_i \left[(I - \Phi_i^\top S_i) (I - \Phi_i^\top S_i)^\top + \beta \Phi_i^\top \Phi_i \right] \mathbf{f}_i^\top \\ &= \mathbf{f}_i L_i \mathbf{f}_i^\top \end{aligned} \quad (13)$$

where

$$L_i = \left[(I - \Phi_i^\top S_i) (I - \Phi_i^\top S_i)^\top + \beta \Phi_i^\top \Phi_i \right] \in R^{k \times k} \quad (14)$$

is a local regularization matrix for learning of \mathbf{f}_i .

Moreover, to consider the summing of the local objective functions of all the data points in (8), we can rewrite \mathbf{f}_i as the product of \mathbf{f} and a nearest neighbor indicator matrix $H_i = \{1, 0\}^{n \times k}$ for each \mathcal{N}_i to indicate which data points are in \mathcal{N}_i . The (j, j') -th element of H_i is defined as

$$H_{ijj'} = \begin{cases} 1, & \mathbf{x}_j \text{ is the } j' - \text{th nearest neighbor of } \mathbf{x}_j, \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

and then \mathbf{f}_i can be rewritten as,

$$\mathbf{f}_i = \mathbf{f} H_i. \quad (16)$$

Substituting both (13) and (16) to (8), the first term of (8) can be rewritten as

$$\begin{aligned} \sum_{i=1}^n g(S_i, \mathbf{f}_i) &= \sum_{i=1}^n \mathbf{f}_i L_i \mathbf{f}_i^\top \\ &= \sum_{i=1}^n \mathbf{f} H_i L_i H_i^\top \mathbf{f}^\top = \mathbf{f} \left(\sum_{i=1}^n H_i L_i H_i^\top \right) \mathbf{f}^\top. \end{aligned} \quad (17)$$

The second term of (8) can also be rewritten in a matrix form as,

$$\begin{aligned} \sum_{i=1}^n \|f_i - y\|_2^2 \lambda_i \\ = (\mathbf{f} - \mathbf{y}) \text{diag}(\boldsymbol{\lambda}) (\mathbf{f} - \mathbf{y})^\top. \end{aligned} \quad (18)$$

where $\text{diag}(\boldsymbol{\lambda}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with its diagonal vector as $\boldsymbol{\lambda}$, and $\mathbf{y} = [y, \dots, y] \in \mathbb{R}^n$ is a n dimensional vector with all its elements as y .

Finally, we substitute (17) and (18) to (8) and obtain the optimization problem with regard to the ranking score vector \mathbf{f} ,

$$\min_{\mathbf{f}} \left\{ h(\mathbf{f}) = \gamma \mathbf{f} \left(\sum_{i=1}^n H_i L_i H_i^\top \right) \mathbf{f}^\top + \delta (\mathbf{f} - \mathbf{y}) \text{diag}(\boldsymbol{\lambda}) (\mathbf{f} - \mathbf{y})^\top \right\}. \quad (19)$$

Where $h(\mathbf{f})$ is the objective function for problem of learning \mathbf{f} . This problem can be easily solved by setting the partial derivative of $h(\mathbf{f})$ with regard to \mathbf{f} to zero,

$$\begin{aligned} \frac{\partial h(\mathbf{f})}{\partial \mathbf{f}} &= 2\gamma \mathbf{f} \left(\sum_{i=1}^n H_i L_i H_i^\top \right) + 2\delta (\mathbf{f} - \mathbf{y}) \text{diag}(\boldsymbol{\lambda}) = 0 \\ \mathbf{f} &= \delta \mathbf{y} \text{diag}(\boldsymbol{\lambda}) \left[\gamma \left(\sum_{i=1}^n H_i L_i H_i^\top \right) + \delta \text{diag}(\boldsymbol{\lambda}) \right]^{-1}. \end{aligned} \quad (20)$$

2.2.2 Solving sparse codes

When the ranking scores $\{f_i\}_{i=1}^n$ and dictionary D is fixed, and the terms irrelevant to sparse codes are removed, the optimization problem in (7) is reduced to,

$$\begin{aligned} \min_{\{\mathbf{s}_i, \mathbf{w}_i\}_{i=1}^n} \sum_{i=1}^n \left(\|\mathbf{x}_i - D\mathbf{s}_i\|_2^2 + \alpha \|\mathbf{s}_i\|_1 \right) \\ + \gamma \sum_{i=1}^n \left(\sum_{j: \mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_i^\top \mathbf{s}_j\|_2^2 + \beta \|\mathbf{w}_i\|_2^2 \right) \end{aligned} \quad (21)$$

Please note that as indicated in (11), the optimal solution of \mathbf{w}_i is also a function of the sparse codes of data points in \mathcal{N}_i . Directly solving this problem will be complex, and choose to use an EM-like algorithm to solve it. In each iteration, \mathbf{w}_i is firstly be estimated using the sparse codes solved in previous iteration, and then it is fixed when the sparse codes $\{\mathbf{s}_i\}_{i=1}^n$ are updated. Moreover, we also choose to update the sparse codes one by one. When the sparse code \mathbf{s}_i is considered, the others $\{\mathbf{s}_j\}_{j: j \neq i}$ are fixed. This reduces the problem in (21) to

$$\begin{aligned} \min_{\mathbf{s}_i} \|\mathbf{x}_i - D\mathbf{s}_i\|_2^2 + \alpha \|\mathbf{s}_i\|_1 \\ + \gamma \sum_{j: \mathbf{x}_j \in \mathcal{N}_i} \|f_j - \mathbf{w}_j^\top \mathbf{s}_i\|_2^2. \end{aligned} \quad (22)$$

This problem can be easily solved by the feature-sign search algorithm [3].

2.2.3 Solving dictionary

When ranking scores and sparse codes are fixed, only the dictionary D is considered, and the irrelevant terms are moved, the problem in (7) is turned to

$$\begin{aligned} \min_D \sum_{i=1}^n \|\mathbf{x}_i - D\mathbf{s}_i\|_2^2 \\ \text{s.t. } \|\mathbf{d}_k\|_2^2 \leq C. \end{aligned} \quad (23)$$

This is a typical dictionary learning problem of sparse coding, and it can be solved by the Lagrange dual method [3].

2.3. Algorithm

Based on the optimization results, we can design an iterative algorithm. The iterations will be repeated until it meets a tolerance stopping criterion or a maximum iteration time is met.

Algorithm 1 Iterative joint sparse coding and ranking score learning algorithm.

Input: Training set of n data points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$;
Input: Query indicator vector λ ;
Input: Tradeoff parameters α, β, γ and δ ;
Input: Tolerance stopping criterion ξ ;
Initialize the sparse codes $\{\mathbf{s}_i^0\}_{i=1}^n$ and the dictionary D^0 ;
Initialize $t = 1$;
Find the k nearest neighbors $\mathcal{N}_i|_{i=1}^n$ and construct the nearest neighbor indicator matrix $H_i|_{i=1}^n$ for the data points $\mathbf{x}_i|_{i=1}^n$.
repeat
 Update $\Phi_i^t|_{i=1}^n$ and $L_i^t|_{i=1}^n$ as in (12) and (14) by fixing the sparse codes as $\{\mathbf{s}_i^{t-1}\}_{i=1}^n$;
 Update the ranking score vector \mathbf{f}^t as in (20);
 Update the local ranking score predictor parameters $\mathbf{w}_i^t|_{i=1}^n$ as in (11) by fixing $\Phi_i^t|_{i=1}^n$ and $\mathbf{f}_i^t|_{i=1}^n$;
 Update the sparse codes $\mathbf{s}_i^t|_{i=1}^n$ one by one by solving (22) by fixing D^{t-1} , $\mathbf{w}_i^t|_{i=1}^n$ and $f_i^t|_{i=1}^n$;
 Update the dictionary D^t by solving (23) by fixing $\mathbf{s}_i^t|_{i=1}^n$;
 $t = t + 1$;
until $\|O(D^{t-1}, \mathbf{s}_i^{t-1}|_{i=1}^n, f_i^{t-1}|_{i=1}^n, \mathbf{w}_i^{t-1}|_{i=1}^n) - O(D^{t-2}, \mathbf{s}_i^{t-2}|_{i=1}^n, f_i^{t-2}|_{i=1}^n, \mathbf{w}_i^{t-2}|_{i=1}^n)\|_1 \leq \xi$ or $t \geq T$
Output: Ranking scores $f_i^{t-1}|_{i=1}^n$, sparse codes $\mathbf{s}_i^{t-1}|_{i=1}^n$ and a dictionary D^{t-1} .

3. Conclusion

In this paper, for the first time, we propose to explore the internal relationship between a popular data representation method, sparse coding, and an important procedure of nearest neighbor search problem, ranking score learning. The connection between them is explored by using a local linear function to approximate the ranking scores from the sparse codes in the local neighborhood of each data point. An unified objective function is constructed based on the local learning of ranking scores from sparse codes, and also based on sparse coding and query information regularization problems. By iteratively optimizing it with regard to the sparse codes, dictionary, and ranking scores, we develop the first joint sparse coding and ranking score learning algorithm.

References

- [1] S. Gao, I. W. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely—laplacian sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3555–3561. IEEE, 2010.
- [2] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 9–16. ACM, 2004.
- [3] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.
- [4] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.
- [5] S. Shekhar, V. Patel, N. Nasrabadi, and R. Chellappa. Joint sparse representation for robust multimodal biometrics recognition. 2014.
- [6] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos. Tensor sparse coding for positive definite matrices. 2013.
- [7] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.
- [8] Y. Yang, D. Xu, F. Nie, J. Luo, and Y. Zhuang. Ranking with local regression and global alignment for cross media retrieval. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 175–184. ACM, 2009.
- [9] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. *Advances in neural information processing systems*, 16:169–176, 2004.
- [10] N. Zhou and J. Fan. Jointly learning visually correlated dictionaries for large-scale visual recognition applications. 2013.